A Linear Lower Bound on the Communication Complexity of Single-Server Private Information Retrieval*

Iftach Haitner, Jonathan J. Hoch, and Gil Segev

Department of Computer Science and Applied Mathematics,
Weizmann Institute of Science, Rehovot 76100, Israel
{iftach.haitner,yaakov.hoch,gil.segev}@weizmann.ac.il

Abstract. We study the communication complexity of single-server Private Information Retrieval (PIR) protocols that are based on fundamental cryptographic primitives in a black-box manner. In this setting, we establish a tight lower bound on the number of bits communicated by the server in any polynomially-preserving construction that relies on trapdoor permutations. More specifically, our main result states that in such constructions $\Omega(n)$ bits must be communicated by the server, where n is the size of the server's database, and this improves the $\Omega(n/\log n)$ lower bound due to Haitner, Hoch, Reingold and Segev (FOCS '07). Therefore, in the setting under consideration, the naive solution in which the user downloads the entire database turns out to be optimal up to constant multiplicative factors. We note that the lower bound we establish holds for the most generic form of trapdoor permutations, including in particular enhanced trapdoor permutations.

Technically speaking, this paper consists of two main contributions from which our lower bound is obtained. First, we derive a tight lower bound on the number of bits communicated by the sender during the commit stage of any black-box construction of a statistically-hiding bit-commitment scheme from a family of trapdoor permutations. This lower bound asymptotically matches the upper bound provided by the scheme of Naor, Ostrovsky, Venkatesan and Yung (CRYPTO '92). Second, we improve the efficiency of the reduction of statistically-hiding commitment schemes to low-communication single-server PIR, due to Beimel, Ishai, Kushilevitz and Malkin (STOC '99). In particular, we present a reduction that essentially preserves the communication complexity of the underlying single-server PIR protocol.

1 Introduction

A single-server Private Information Retrieval (PIR) scheme is a protocol between a server and a user. The server holds a database $x \in \{0,1\}^n$ and the user holds an index $i \in [n]$ to an entry of the database. Informally, the user wishes to retrieve the i^{th} entry of the database, without revealing the index i to the server. The

[*] Due to space limitations a more complete version is available as [19].

R. Canetti (Ed.): TCC 2008, LNCS 4948, pp. 445–464, 2008.

notion of PIR was introduced by Chor, Goldreich, Kushilevitz and Sudan [4] to model applications that enable users to query public databases without revealing any information on the specific data that the users wish to retrieve. Chor et al. showed that in the information-theoretic setting any single-server PIR protocol has the server communicating at least n bits. Therefore in this setting the naive solution in which the user downloads the entire database is optimal.

Kushilevitz and Ostrovsky [26] were the first to construct a non-trivial single-server PIR protocol relying on computational assumptions. Their result initiated a sequence of papers showing that there exist single-server PIR protocols with poly-logarithmic communication complexity based on *specific* number-theoretic assumptions (see, for example, [2,3,12,26,28,40], and a recent survey by Ostrovsky and Skeith [35]). The only non-trivial construction based on *general* computational assumptions is due to Kushilevitz and Ostrovsky [27], and is based on enhanced trapdoor permutations. In their construction, however, the server is required to communicate $n - o(n)$ bits to the user.

Motivated by this ever-growing line of work, we study the communication complexity of single-server PIR protocols that are based on fundamental primitives. We establish a linear lower bound on the number of bits communicated by the server in constructions that rely on enhanced trapdoor permutations in a black-box manner. Therefore, in the setting under consideration in this paper, the naive solution in which the user downloads the entire database turns out to be optimal up to constant multiplicative factors. In the following paragraphs, we briefly describe the setting in which our lower bound is proved (a more formal description is provided in Section 2).

Black-box reductions. As previously mentioned, under widely believed specific number-theoretic assumptions, there are very efficient single-server PIR protocols. Therefore, if any of these assumptions holds, the existence of trapdoor permutations implies the existence of efficient single-server PIR protocols in a trivial sense. Faced with similar difficulties, Impagliazzo and Rudich [22] presented a paradigm for proving impossibility results under a restricted, yet very natural and important, subclass of reductions called *black-box reductions*. Informally, a black-box reduction of a primitive P to a primitive Q is a construction of P out of Q that ignores the internal structure of the implementation of Q and uses it as a "subroutine" (i.e., as a black-box). In addition, in the case of fully-black-box reductions (see, for example, [36]), the proof of security (showing that an adversary that breaks the implementation of P implies an adversary that breaks the implementation of Q), is black-box as well, that is, the internal structure of the adversary that breaks the implementation of P is ignored.

The strength of cryptographic reductions. Luby [30] provides a classification of the strength of cryptographic reductions into three classes: linearly-preserving, polynomially-preserving and weakly-preserving. In our setting, this classification comes into play when comparing the size of the server's database and the domain of the trapdoor permutations. Very informally, a reduction of single-server PIR for an n-bit database to a family of trapdoor permutations is linearly-preserving or polynomially-preserving if it uses trapdoor permutations over $\Omega(n)$ bits. Such a reduction is weakly-preserving if it uses trapdoor permutations over $\Omega(n^{\epsilon})$ bits for some constant $0 < \epsilon \leq 1$. In linearly-preserving and polynomially-preserving reductions we are guaranteed that breaking the constructed primitive is essentially as hard as breaking the underlying primitive. However, in weakly-preserving reductions, we are only guaranteed that breaking the constructed primitive is as hard as breaking the underlying primitive for polynomially smaller security parameters. We refer the reader to [30] for a more comprehensive and complete discussion.

1.1 Related Work

Single-server PIR is one of the fundamental primitives in the foundations of cryptography. For example, non-trivial single-server PIR was shown to imply the existence of Oblivious Transfer protocols [5], and 2-move low-communication single-server PIR was shown to imply collision-resistant hash functions [23]. Single-server PIR was also shown to be tightly related to several other aspects of cryptography and complexity theory (see, for example, [6,20,24]). We note that it is far beyond the scope of this paper to present an exhaustive overview of the ever-growing line of work on single-server PIR, and we refer the reader to the recent survey of Ostrovsky and Skeith [35] for a more comprehensive discussion.

In the context of black-box reductions, Impagliazzo and Rudich [22] showed that there are no black-box reductions of key-agrement protocols to one-way permutations, and substantial additional work in this line followed (see, for example, [7,13,14,37,38]). Kim, Simon and Tetali [25] initiated a new line of impossibility results, by providing a lower bound on the *efficiency* of black-box reductions (rather than on their feasibility). They proved a lower bound on the efficiency, in terms of the number of calls to the underlying primitive, of any black-box reduction of universal one-way hash functions to one-way permutations. This result was later improved, to match the known upper bound, by Gennaro and Trevisan [11], which together with Gennaro et al. [8,9] provided tight lower bounds on the efficiency of several other black-box reductions. Building upon the technique developed by [11], Horvitz and Katz [21] provided lower bounds on the efficiency of black-box reductions of statistically-hiding and computationally-binding commitment schemes to one-way permutations. In the above results the measure of efficiency under consideration is the number of calls to the underlying primitives.

Di Crescenzo, Malkin and Ostrovsky [5] showed that any single-server PIR protocol in which the server communicates at most $n-1$ bits (where n is the size of the server's database) can be transformed in a fully-black-box manner to an Oblivious Transfer protocol. Gennaro, Lindell and Malkin [10] (refining Gertner et al. [13]) ruled out any black-box reduction of Oblivious Transfer to plain (i.e., non-enhanced) trapdoor permutations. The combination of these two results yields that there are no non-trivial black-box constructions of single-server PIR from non-enhanced trapdoor permutations. We note that although in this paper we rule out a more restricted class of constructions (that is, the class of fully-black-box constructions), our result holds for the most generic form of trapdoor permutations, including in particular enhanced trapdoor permutations.

Very recently, Haitner et al. [18], improving upon the work of Wee [41], proved that any polynomially-preserving fully-black-box reduction of a statistically-hiding bit-commitment scheme to trapdoor permutations has $\Omega(n/\log n)$ communication rounds (where n is the security parameter). As a corollary, they showed that any polynomially-preserving fully-black-box reduction of single-server PIR to trapdoor permutations has $\Omega(n/\log n)$ communication rounds, where n is the size of the server's database. In particular, the server is required to communicate $\Omega(n/\log n)$ bits to the user. Haitner et al. also established similar lower bounds on the communication complexity of Oblivious Transfer that guarantees statistical security for one of the parties and for Interactive Hashing.

In a slightly different setting, Ostrovsky and Skeith [34] proved a lower bound on the communication complexity of single-server PIR protocols with certain algebraic properties. For a class of PIR protocols, referred to as *abelian group algebraic PIR protocols*, with user-side communication complexity $g(n)$ and server-side communication complexity $h(n)$ they proved that $g(n)h(n) = \Omega(n)$.

1.2 Our Results

We study the class of black-box constructions of single-server PIR from trapdoor permutations, and establish a tight lower bound on the number of bits communicated by the server in such constructions. Our main result is the following:

Main Theorem (Informal). *In any polynomially-preserving fully-black-box construction of a single-server PIR protocol from trapdoor permutations the server communicates $\Omega(n)$ bits, where n is the size of the server's database.*

As mentioned above, the combination of the results of Di Crescenzo et al. [5] and of Gennaro et al. [10] rules out the more general class of black-box reductions of single-server PIR with $n-1$ bits of communication to trapdoor permutations. This result, however, does not apply to enhanced trapdoor permutations. We note that our lower bound holds for the most generic form of trapdoor permutations, and in particular for enhanced trapdoor permutations.[1]

In addition, we note that our lower bound holds only for constructions which are polynomially-preserving. The construction of Kushilevitz and Ostrovsky [27], which is based on enhanced trapdoor permutations in a fully-black-box manner and in which the server communicates $n - o(n)$ bits, is only weakly-preserving (i.e., it is significantly easier to break their protocol than to break the security of the underlying family of trapdoor permutations [2]). Thus, the question of whether a tight linear lower bound can be established for weakly-preserving constructions as well remains open.

[1] Note that enhanced trapdoor permutations are, seemingly, stronger than plain trapdoor permutations. Therefore, although our result is weaker in terms of the class of reductions and the bound on the communication complexity, it provides the first evidence that enhanced trapdoor permutations are not sufficient to construct single-server PIR with sublinear communication (at least from a black-box perspective).

[2] Though the security guarantees of the two primitives are still polynomially-related.

The main technical contributions. This paper consists of two main contributions from which our lower bound is immediately obtained. First, we derive a tight lower bound on the communication complexity of black-box constructions of statistically-hiding bit-commitment schemes from trapdoor permutations. Very recently, Haitner et al. [18] proved that any polynomially-preserving fully-black-box construction of statistically-hiding bit-commitment scheme from a family of trapdoor permutations has $\Omega(n/\log n)$ communication rounds, where n is the security parameter of the scheme. In particular, this implies a lower bound on the number of bits communicated by the sender. In this paper we manage to improve their lower bound and prove the following theorem:

Theorem (Informal) 1.1. *In any polynomially-preserving fully-black-box construction of a statistically-hiding bit-commitment scheme from a family of trapdoor permutations the sender communicates $\Omega(n)$ bits during the commit stage, where n is the security parameter of the scheme.*

This lower bound asymptotically matches the upper bound given by the statistically-hiding commitment scheme of Naor et al. [31]. In addition, we improve the efficiency of the reduction of statistically-hiding commitment schemes to single-server PIR, presented by Beimel et al. [1]. Our reduction essentially uses the reduction of Beimel et al. instantiated with a better extractor, which enables us to preserve the communication complexity of the underlying single-server PIR protocol. As stating this result turns out to involve subtle technical details, here we only state a very informal statement:

Theorem (Informal) 1.2. *There is a linearly-preserving fully-black-box reduction of statistically-hiding commitment schemes to low-communication single-server PIR, which essentially preserves the communication complexity of the underlying single-server PIR protocol.*

Paper organization. In Section 2 we briefly present the notations and formal definitions used in this paper. In Section 3 we prove a tight lower bound on the number of bits communicated by the sender during the commit stage of statistically-hiding commitment schemes. In Section 4 we describe an improved reduction of statistically-hiding commitment schemes to single-server PIR. Finally, in Section 5 we provide some concluding remarks.

2 Preliminaries

We denote by Π_n the set of all permutations over $\{0,1\}^n$. For an integer n, we denote by U_n the uniform distribution over the set $\{0,1\}^n$. For a finite set X, we denote by $x \leftarrow X$ the experiment of choosing an element of X according to the uniform distribution. Similarly, for a distribution $\mathcal{D}$ over a set X, we denote by $x \leftarrow \mathcal{D}$ the experiment of choosing an element of X according to the distribution $\mathcal{D}$. The min-entropy of $\mathcal{D}$ is defined as $\mathrm{H}_\infty(\mathcal{D}) = -\log\left(\max_x \Pr_{\mathcal{D}}[x]\right)$. The statistical distance between two distributions X and Y over Ω is defined as $\mathrm{SD}(X,Y) = \frac{1}{2}\sum_{\omega\in\Omega} \left|\Pr_X[\omega] - \Pr_Y[\omega]\right|$.

Definition 2.1. *A function $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a (k,ϵ)-extractor if for every distribution X over $\{0,1\}^n$ with $\mathrm{H}_\infty(X) \geq k$, it holds that the distribution $E(X, U_d)$ is ϵ-close to uniform. Such a function E is a* strong (k,ϵ)-*extractor if the function $E'(x,y) = y \circ E(x,y)$ is a (k,ϵ)-extractor (where $\circ$ denotes concatenation).*

In our construction of a statistically-hiding commitment scheme from single-server PIR we will be using the following explicit construction of strong extractors, which is obtained as a corollary of [39, Corollary 3.4].

Proposition 2.1. *For any $k \in \omega(\log(n))$, there exists an explicit construction of a strong $(k, 2^{1-k})$-extractor* $\mathrm{EXT} : \{0,1\}^n \times \{0,1\}^{3k} \to \{0,1\}^{k/2}$.

Trapdoor permutations. We briefly present the notion of trapdoor permutations, and refer the reader to [15] for a more comprehensive discussion. A collection of trapdoor permutations is represented by a triplet of the form $\tau = \left(G, F, F^{-1}\right)$. Informally, G corresponds to a key generation procedure, which is queried on a string td (intended as the "trapdoor") and produces a corresponding public key pk. The procedure F is the actual collection of permutations, which is queried on a public key pk and an input x. Finally, the procedure F^{-1} is the inverse of F: If $G(td) = pk$ and $F(pk, x) = y$, then $F^{-1}(td, y) = x$. In this paper, since we are concerned with providing a lower bound, we do not consider the most general definition of a collection of trapdoor permutations. Instead, we denote by T_n the set of all triplets $\tau_n = \left(G_n, F_n, F_n^{-1}\right)$ of the following form:

1. $G_n \in \Pi_n$.
2. $F_n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ is a function such that $F_n(pk, \cdot) \in \Pi_n$ for every $pk \in \{0,1\}^n$.
3. $F_n^{-1} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ is a function such that $F_n^{-1}(td, y)$ returns the unique $x \in \{0,1\}^n$ for which $F_n(G_n(td), x) = y$.

Our lower bound proof is based on analyzing random instances of such collections. A uniformly distributed $\tau_n \in T_n$ can be chosen as follows: G_n is chosen uniformly at random from Π_n, and for each $pk \in \{0,1\}^n$ a permutation $F_n(pk, \cdot)$ is chosen uniformly and independently at random from Π_n.

Definition 2.2. *A family $\tau = \left\{\tau_n = \left(G_n, F_n, F_n^{-1}\right)\right\}_{n=1}^{\infty}$ of trapdoor permutations is $s(n)$-*hard *if for every probabilistic Turing-machine A that runs in time $s(n)$, and for all sufficiently large n,*

$$\Pr\left[A^{\tau}(1^n, G_n(td), y) = F_n^{-1}(td, y)\right] \leq \frac{1}{s(n)} \ ,$$

where the probability is taken uniformly over all the possible choices of $td \in \{0,1\}^n$ and $y \in \{0,1\}^n$, and over all the possible outcomes of the internal coin tosses of A.

Definition 2.2 refers to the difficulty of inverting a random permutation $F(pk, \cdot)$ on a uniformly distributed image y, when given only $pk = G(td)$ and y.

Some applications, however, require enhanced hardness conditions. For example, it may be required (cf. [16, Appendix C]) that it is hard to invert $F(pk, \cdot)$ on y even given the random coins used in the generation of y. Note that our formulation captures such hardness condition as well and therefore the impossibility results proved in this paper hold also for enhanced trapdoor permutations.[3]

Single-server Private Information Retrieval. A single-server Private Information Retrieval (PIR) scheme is a protocol between a server and a user. The server holds a database $x \in \{0,1\}^n$ and the user holds an index $i \in [n]$ to an entry of the database. Very informally, the user wishes to retrieve the i^{th} entry of the database, without revealing the index i to the server. More formally, a single-server PIR scheme is defined via a pair of probabilistic polynomial-time Turing-machines $(\mathcal{S},\mathcal{U})$ such that:

- $\mathcal{S}$ receives as input a string $x \in \{0,1\}^n$. Following its interaction it does not have any output.
- $\mathcal{U}$ receives as input an index $i \in [n]$. Following its interaction it outputs a value $b \in \{0, 1, \perp\}$.

Denote by $b \leftarrow \langle \mathcal{S}(x),\mathcal{U}(i)\rangle$ the experiment in which $\mathcal{S}$ and $\mathcal{U}$ interact (using the given inputs and uniformly chosen random coins), and then $\mathcal{U}$ outputs the value b. It is required that there exists a negligible function $\nu(n)$, such that for all sufficiently large n, and for every string $x = x_1 \circ \cdots \circ x_n \in \{0,1\}^n$, it holds that $x_i \leftarrow \langle \mathcal{S}(x),\mathcal{U}(i)\rangle$ with probability at least $1 - \nu(n)$ over the random coins of both $\mathcal{S}$ and $\mathcal{R}$.

In order to define the security properties of such schemes, we first introduce the following notation. Given a single-server PIR scheme $(\mathcal{S},\mathcal{U})$ and a Turing-machine $\mathcal{S}^*$ (a malicious server), we denote by $\mathsf{view}_{\langle \mathcal{S}^*,\mathcal{U}(i)\rangle}(n)$ the distribution on the view of $\mathcal{S}^*$ when interacting with $\mathcal{U}(i)$ where $i \in [n]$. This view consists of its random coins and of the sequence of messages it receives from $\mathcal{U}$, where the distribution is taken over the random coins of both $\mathcal{S}^*$ and $\mathcal{U}$.

Definition 2.3. *A single-server PIR scheme $(\mathcal{S},\mathcal{U})$ is secure if for every probabilistic polynomial-time Turing-machines $\mathcal{S}^*$ and $\mathcal{D}$, and for every two sequences of indices $\{i_n\}_{i=1}^{\infty}$ and $\{j_n\}_{i=1}^{\infty}$ where $i_n, j_n \in [n]$ for every n, it holds that*

$$\Big|\Pr\left[v \leftarrow \mathsf{view}_{\langle \mathcal{S}^*,\mathcal{U}(i_n)\rangle}(n) : \mathcal{D}(v) = 1\right] \\ -\Pr\left[v \leftarrow \mathsf{view}_{\langle \mathcal{S}^*,\mathcal{U}(j_n)\rangle}(n) : \mathcal{D}(v) = 1\right]\Big| \leq \nu(n) \ ,$$

for some negligible function $\nu(n)$ and for all sufficiently large n.

Commitment schemes. A commitment scheme is a two-stage interactive protocol between a sender and a receiver. Informally, after the first stage of the protocol, which is referred to as the *commit stage*, the sender is bound to at

[3] A different enhancement, used by [17], requires the permutations' domain to be polynomially dense in $\{0,1\}^n$. Clearly, our impossibility result holds for such an enhancement as well.

most one value, not yet revealed to the receiver. In the second stage, which is referred to as the *reveal stage*, the sender reveals its committed value to the receiver. More formally, a commitment scheme is defined via a triplet of probabilistic polynomial-time Turing-machines $(\mathcal{S}, \mathcal{R}, \mathcal{V})$ such that:

- $\mathcal{S}$ receives as input the security parameter 1^n and a string $x \in \{0,1\}^k$. Following its interaction, it outputs some information decom (the decommitment).
- $\mathcal{R}$ receives as input the security parameter 1^n. Following its interaction, it outputs a state information com (the commitment).
- $\mathcal{V}$ (acting as the receiver in the reveal stage[4]) receives as input the security parameter 1^n, a commitment com and a decommitment decom. It outputs either a string $x' \in \{0,1\}^k$ or $\perp$.

Denote by $(\mathsf{decom}|\mathsf{com}) \leftarrow \langle \mathcal{S}(1^n, x), \mathcal{R}(1^n) \rangle$ the experiment in which $\mathcal{S}$ and $\mathcal{R}$ interact (using the given inputs and uniformly chosen random coins), and then $\mathcal{S}$ outputs decom while $\mathcal{R}$ outputs com. It is required that for all n, every string $x \in \{0,1\}^k$, and every pair $(\mathsf{decom}|\mathsf{com})$ that may be output by $\langle \mathcal{S}(1^n, x), \mathcal{R}(1^n) \rangle$, it holds that $\mathcal{V}(\mathsf{com}, \mathsf{decom}) = x$.[5] In the remainder of the paper, it will often be convenient for us to identify $\mathcal{V}$ with $\mathcal{R}$, and refer to a commitment scheme as a pair $(\mathcal{S}, \mathcal{R})$.

The security of a commitment scheme can be defined in two complementary ways, protecting against either an all-powerful sender or an all-powerful receiver. In this paper, we deal with commitment schemes of the latter type, which are referred to as *statistically-hiding* commitment schemes. In order to define the security properties of such schemes, we first introduce the following notation. Given a commitment scheme $(\mathcal{S}, \mathcal{R})$ and a Turing-machine $\mathcal{R}^*$, we denote by $\mathsf{view}_{\langle \mathcal{S}(x), \mathcal{R}^* \rangle}(n)$ the distribution on the view of $\mathcal{R}^*$ when interacting with $\mathcal{S}(1^n, x)$. This view consists of $\mathcal{R}^*$'s random coins and of the sequence of messages it receives from $\mathcal{S}$. The distribution is taken over the random coins of both $\mathcal{S}$ and $\mathcal{R}^*$. Note that whenever no computational restrictions are assumed on $\mathcal{R}^*$, without loss of generality we can assume that $\mathcal{R}^*$ is deterministic.

Definition 2.4. *A commitment scheme $(\mathcal{S}, \mathcal{R})$ is $\rho(n)$-*hiding *if for every deterministic Turing-machine $\mathcal{R}^*$, and for every two sequences of strings $\{x_n\}_{i=1}^{\infty}$ and $\{x'_n\}_{i=1}^{\infty}$ where $x_n, x'_n \in \{0,1\}^{k(n)}$ for every n the ensembles $\{\mathsf{view}_{\langle \mathcal{S}(x_n), \mathcal{R}^* \rangle}(n)\}$ and $\{\mathsf{view}_{\langle \mathcal{S}(x'_n), \mathcal{R}^* \rangle}(n)\}$ have statistical difference at most $\rho(n)$ for all sufficiently large n. Such a scheme is* statistically-hiding *if it is $\rho(n)$-hiding for some negligible function $\rho(n)$.*

Our lower bound for commitment schemes holds in fact under a weaker hiding requirement. We derive our results even for commitment schemes in which the

[4] Note that there is no loss of generality in assuming that the reveal stage is non-interactive. This is since any such interactive stage can be replaced with a non-interactive one as follows: The sender sends its internal state to the receiver, who then simulates the sender in the interactive stage.

[5] Although we assume perfect completeness, it is not essential for our results.

sender is statistically protected only against an honest receiver. Such schemes are referred to as *statistically-hiding honest-receiver* commitment schemes. Formally, it is only required that the statistical difference between the ensembles $\{\mathsf{view}_{\langle \mathcal{S}(x_n),\mathcal{R}\rangle}(n)\}$ and $\{\mathsf{view}_{\langle \mathcal{S}(x'_n),\mathcal{R}\rangle}(n)\}$ is some negligible function of n.

Definition 2.5. *A commitment scheme* $(\mathcal{S},\mathcal{R},\mathcal{V})$ *is* $\mu(n)$-binding *if for every probabilistic polynomial-time Turing-machine* $\mathcal{S}^*$ *it holds that the probability that* $((\mathsf{decom},\mathsf{decom}')|\mathsf{com}) \leftarrow \langle \mathcal{S}^*(1^n),\mathcal{R}(1^n)\rangle$ *(where the probability is over the random coins of both* $\mathcal{S}^*$ *and* $\mathcal{R}$*) such that* $\mathcal{V}(\mathsf{com},\mathsf{decom}) \neq \mathcal{V}(\mathsf{com},\mathsf{decom}')$ *and* $\mathcal{V}(\mathsf{com},\mathsf{decom}),\mathcal{V}(\mathsf{com},\mathsf{decom}') \neq \perp$ *is negligible in* n *for all sufficiently large* n*. Such a scheme is* computationally-binding *if it is* $\mu(n)$*-binding for some negligible function* $\mu(n)$*, and is* weakly-binding *if it is* $(1-1/p(n))$*-binding for some polynomial* $p(n)$*.*

Black-box reductions. A reduction of a primitive P to a primitive Q is a construction of P out of Q. Such a construction consists of showing that if there exists an implementation C of Q, then there exists an implementation M_C of P. This is equivalent to showing that for every adversary that breaks M_C, there exists an adversary that breaks C. Such a reduction is *semi-black-box* if it ignores the internal structure of Q's implementation, and it is *fully-black-box* if the proof of correctness is black-box as well, i.e., the adversary for breaking Q ignores the internal structure of both Q's implementation and of the (alleged) adversary breaking P. Semi-black-box reductions are less restricted and thus more powerful than fully-black-box reductions. A taxonomy of black-box reductions was provided by Reingold, Trevisan and Vadhan [36], and the reader is referred to their paper for a more complete and formal view of these notions.

We now formally define the class of constructions considered in this paper. Our results in the current paper are concerned with the particular setting of fully-black-box constructions of single-server PIR and of statistically-hiding commitment schemes from trapdoor permutations. We focus here on specific definitions for these particular primitives and we refer the reader to [36] for a more general definition.

When examining efficiency measures of fully-black-box constructions, an essential parameter for such characterizations, as introduced by Haitner et al. [18], is the *security-parameter-expansion* of the construction. Consider, for example, a fully-black-construction of a commitment scheme from a family of trapdoor permutations. One ingredient of such a construction is a machine A that attempts to break the security of the trapdoor permutation family given oracle access to any malicious sender $\mathcal{S}^*$ that breaks the security of the commitment scheme. Then, A receives a security parameter 1^n (and possibly some additional inputs) and invokes $\mathcal{S}^*$ in a black-box manner. The standard definition does not restrict the range of security parameters that A is allowed to invoke $\mathcal{S}^*$ on. For example, A may invoke $\mathcal{S}^*$ on security parameter 1^{n^2}, or even on security parameter $1^{\Theta(s(n))}$, where $s(n)$ is the running time of A. In this paper, we will use the notion $\ell(n)$-expanding for short, and note that according to Luby's classification [30], any polynomially-preserving reduction is $O(n)$-expanding in our terminology.

Definition 2.6. *A fully-black-box $\ell(n)$-expanding construction of a single-server PIR scheme from an $s(n)$-hard family of trapdoor permutations is a triplet of probabilistic oracle Turing-machines $(\mathcal{S},\mathcal{U},A)$ for which the following hold:*

1. **Correctness:** *For every family τ of trapdoor permutations, $(\mathcal{S}^\tau,\mathcal{U}^\tau)$ is a single-server PIR scheme.*
2. **Black-box proof of security:** *For every family of trapdoor permutations $\tau = \left\{\tau_n = \left(G_n, F_n, F_n^{-1}\right)\right\}_{n=1}^{\infty}$ and for every probabilistic polynomial-time Turing-machine $\mathcal{S}^*$, if $\mathcal{S}^*$ with oracle access to τ breaks the security of $(\mathcal{S}^\tau,\mathcal{U}^\tau)$, then*

$$\Pr\left[A^{\tau,\mathcal{S}^*}(1^n, G_n(td), y) = F_n^{-1}(td, y)\right] > \frac{1}{s(n)} \ ,$$

for infinitely many values of n, where A runs in time $s(n)$ and invokes $\mathcal{S}^*$ on security parameters which are at most $1^{\ell(n)}$. The probability is taken uniformly over all the possible choices of $td \in \{0,1\}^n$ and $y \in \{0,1\}^n$, and over all the possible outcomes of the internal coin tosses of A.

Definition 2.7. *A fully-black-box $\ell(n)$-expanding construction of a statistically-hiding (against an honest-receiver) and weakly-binding commitment scheme from an $s(n)$-hard family of trapdoor permutations is a triplet of probabilistic oracle Turing-machines $(\mathcal{S},\mathcal{R},A)$ for which the following hold:*

1. **Correctness:** *For every family τ of trapdoor permutations, $(\mathcal{S}^\tau,\mathcal{R}^\tau)$ is a statistically-hiding honest-receiver commitment scheme.*
2. **Black-box proof of binding:** *For every family of trapdoor permutations $\tau = \left\{\tau_n = \left(G_n, F_n, F_n^{-1}\right)\right\}_{n=1}^{\infty}$ and for every probabilistic polynomial-time Turing-machine $\mathcal{S}^*$, if $\mathcal{S}^*$ with oracle access to τ breaks the binding of $(\mathcal{S}^\tau,\mathcal{R}^\tau)$, then*

$$\Pr\left[A^{\tau,\mathcal{S}^*}(1^n, G_n(td), y) = F_n^{-1}(td, y)\right] > \frac{1}{s(n)} \ ,$$

for infinitely many values of n, where A runs in time $s(n)$ and invokes $\mathcal{S}^*$ on security parameters which are at most $1^{\ell(n)}$. The probability is taken uniformly over all the possible choices of $td \in \{0,1\}^n$ and $y \in \{0,1\}^n$, and over all the possible outcomes of the internal coin tosses of A.

3 Communication Lower Bound for Statistically-Hiding Commitment Schemes

In this section we prove a lower bound on the communication complexity of fully-black-box constructions of statistically-hiding commitment schemes from trapdoor permutations. We establish a lower bound on the number of bits communicated by the sender during the commit stage of any such scheme. Since we are interested in proving an impossibility result for commitment schemes, it will be sufficient for us to deal with bit-commitment schemes. We prove the following theorem:

Theorem 3.1. *In any fully-black-box $O(n)$-expanding construction of a weakly-binding statistically-hiding honest-receiver bit-commitment scheme from a family of trapdoor permutations, the sender communicates $\Omega(n)$ bits during the commit stage.*

The proof of Theorem 3.1 follows the approach and technique of Haitner at el. [18] who constructed a "collision-finding" oracle in order to derive a lower bound on the round complexity of statistically-hiding commitment schemes. Given any fully-black-box $O(n)$-expanding construction $(\mathcal{S}, \mathcal{R}, A)$ of a weakly-binding statistically-hiding honest-receiver bit-commitment scheme from a family of trapdoor permutations τ, we show that relative to their oracle the following holds: (1) there exists a malicious sender $\mathcal{S}^*$ that breaks the binding of the scheme $(\mathcal{S}^\tau, \mathcal{R}^\tau)$, and (2) if the sender communicates $o(n)$ bits during the commit stage of $(\mathcal{S}^\tau, \mathcal{R}^\tau)$, then the machine A (with oracle access to $\mathcal{S}^*$) fails to break the security of τ.

3.1 The Oracle

We briefly describe the oracle constructed by Haitner et al. [18] and state its main property. The oracle is of the form $\mathcal{O} = (\tau, \mathsf{Sam}^\tau)$, where τ is a family of trapdoor permutations (i.e., $\tau = \{\tau_n\}_{n=1}^{\infty}$, where $\tau_n \in T_n$ for every n), and Sam^τ is an oracle that, very informally, receives as input a description of a circuit C (which may contain τ-gates) and a string z, and outputs a uniformly distributed preimage of z under the mapping defined by C. As discussed in [18], several essential restrictions are imposed on the querying of Sam that prevent it from assisting in inverting τ.

Description of Sam. The oracle Sam receives as input a query of the form $Q = (C^\tau_{\text{next}}, C^\tau, z)$, and outputs a pair (w', z') where w' is a uniformly distributed preimage of z under the mapping defined by the circuit C^τ, and $z' = C^\tau_{\text{next}}(w')$. We impose the following restrictions:

1. z was the result of a previous query with C^τ as the next-query circuit (note that this imposes a forest-like structure on the queries).
2. The circuit C^τ_{next} is a *refinement* of the circuit C^τ, where by a refinement we mean that $C^\tau_{\text{next}}(w) = (C^\tau(w), \widetilde{C}^\tau(w))$ for some circuit $\widetilde{C}^\tau$ and for every w. In particular, this implies that C^τ and C^τ_{next} have the same input length. Given a query Q, we denote this input length by $m(Q)$, and when the query Q is clear from the context we will write only m.
3. Each query contains a security parameter 1^n, and Sam answers queries only up to depth $\mathsf{depth}(n)$, for some "depth restriction" function $\mathsf{depth} : \mathbb{N} \to \mathbb{N}$ which is a part of the description of Sam. The security parameter is set such that a query with security parameter 1^n is allowed to contain circuits with queries to permutations on up to n bits. Note that although different queries may have different security parameters, we ask that in the same "query-tree", all queries will have the same security parameter (hence the depth of the tree is already determined by the root query).

In order to impose these restrictions, Sam is equipped with a family $\mathsf{sign} = \{\mathsf{sign}_k\}_{k=1}^{\infty}$ of (random) functions $\mathsf{sign}_k : \{0,1\}^k \to \{0,1\}^{2k}$ that will be used as "signatures" for identifying legal queries as follows: in addition to outputting (w', z'), Sam will also output the value $\mathsf{sign}(1^n, C^\tau_{\text{next}}, z', dep + 1)$, where dep is the depth of the query, 1^n is the security parameter of the query, and by applying the "function" sign we actually mean that we apply the function sign_k for the correct input length. Each query of the form $Q = (1^n, C^\tau_{\text{next}}, C^\tau, z, dep, sig)$ is answered by Sam if and only if C^τ_{next} is a refinement of C^τ, $dep \leq \mathsf{depth}(n)$ and $sig = \mathsf{sign}(1^n, C^\tau, z, dep)$.

Finally, Sam is provided with a family of (random) permutations $\mathcal{F} = \{f_Q\}$, where for every possible query Q a permutation f_Q is chosen uniformly at random from $\Pi_{m(Q)}$. Given a query $Q = (1^n, C^\tau_{\text{next}}, C^\tau, z, dep, sig)$, the oracle Sam uses the permutation $f_Q \in \mathcal{F}$ in order to sample w' as follows: it outputs $w' = f_Q(t)$ for the lexicographically smallest $t \in \{0,1\}^m$ such that $C^\tau(f_Q(t)) = z$. Note that whenever the permutation f_Q is chosen from Π_m uniformly at random, and independently of all other permutations in $\mathcal{F}$, then w' is indeed a uniformly distributed preimage of z. In this paper, whenever we consider the probability of an event over the choice of the family $\mathcal{F}$, we mean that for each query Q a permutation f_Q is chosen uniformly at random from $\Pi_{m(Q)}$ and independently of all other permutations. A complete and formal description of the oracle is provided in Figure 3.1.

```
On input $Q = (1^n, C^\tau_{\text{next}}, C^\tau, z, dep, sig)$, $\mathsf{Sam}^{\tau,\mathcal{F},\mathsf{sign}}_{\mathsf{depth}}$ acts as follows:
  1. If $C^\tau = \perp$, then output $(w', z', sig')$ where $w' = f_Q(0^m)$, $z' = C^\tau_{\text{next}}(w')$, and $sig' = \mathsf{sign}(1^n, C^\tau_{\text{next}}, z', 1)$.
  2. Else, if $C^\tau_{\text{next}}$ is a refinement of $C^\tau$, $dep \leq \mathsf{depth}(n)$ and $sig = \mathsf{sign}(1^n, C^\tau, z, dep)$, then
     (a) Find the lexicographically smallest $t \in \{0,1\}^m$ such that $C^\tau(f_Q(t)) = z$.
     (b) Output $(w', z', sig')$ where $w' = f_Q(t)$, $z' = C^\tau_{\text{next}}(w')$, and $sig' = \mathsf{sign}(1^n, C^\tau_{\text{next}}, z', dep + 1)$.
  3. Else, output $\perp$.
```

Fig. 1. The oracle Sam

Definition 3.1. *We say that a circuit A queries the oracle $\mathsf{Sam}^{\tau,\mathcal{F},\mathsf{sign}}_{\mathsf{depth}}$ up to depth d, if for every Sam-query $Q = (1^n, C^\pi_{\text{next}}, C^\pi, z, dep, sig)$ that A makes, it holds that $dep \leq d$.*

One of the main properties of the oracle Sam, as proved in [18], is the following: any circuit with oracle access to Sam that tries to invert a random trapdoor permutation, fails with high probability. More specifically, Haitner et al. managed to relate this success probability to the maximal depth of the Sam-queries made by the circuit, and to the size of the circuit. They proved the following theorem:

Theorem 3.2 ([18]). *For every circuit A of size $s(n)$ that queries* Sam *up to depth $d(n)$ such that $s(n)^{3d(n)+2} < 2^{n/8}$, for every depth restriction function* depth *and for all sufficiently large n, it holds that*

$$\Pr_{\substack{td \leftarrow \{0,1\}^n, \tau, \mathcal{F} \\ y \leftarrow \{0,1\}^n, \mathsf{sign}}} \left[A^{\tau, \mathsf{Sam}_{\mathsf{depth}}^{\tau,\mathcal{F},\mathsf{sign}}}(G_n(td), y) = F_n^{-1}(td, y) \right] \leq \frac{2}{s(n)} \ .$$

3.2 Breaking Low-Communication Statistically-Hiding Commitment Schemes

We show that a random instance of the oracle Sam can be used to break the binding of any statistically-hiding commitment scheme. Specifically, for every bit-commitment scheme $(\mathcal{S}, \mathcal{R})$ which is (1) weakly-biding, (2) statistically-hiding against an honest-receiver, and (3) has oracle access to a family τ of trapdoor permutations, we construct a malicious sender $\mathcal{S}^*$ which has oracle access to $\mathsf{Sam}_{\mathsf{depth}}^{\tau,\mathcal{F},\mathsf{sign}}$, and breaks the binding of $(\mathcal{S}^\tau, \mathcal{R}^\tau)$ with sufficiently high probability over the choices of τ, $\mathcal{F}$ and sign. Formally, the following theorem is proved:

Theorem 3.3. *For any statistically-hiding bit-commitment scheme $(\mathcal{S}, \mathcal{R}, \mathcal{V})$ with oracle access to a family of trapdoor permutations in which the sender communicates at most $c(n)$ bits during the commit stage, and for any polynomial $p(n)$, there exists a polynomial-time malicious sender $\mathcal{S}^*$ such that*

$$\Pr_{\substack{\tau, \mathcal{F} \\ \mathsf{sign}, r_{\mathcal{R}}}} \left[\begin{array}{r} ((\mathsf{decom}, \mathsf{decom}')|\mathsf{com}) \leftarrow \left\langle \mathcal{S}^{*\ \mathsf{Sam}_{\mathsf{depth}}^{\tau,\mathcal{F},\mathsf{sign}}}(1^n), \mathcal{R}^\tau(1^n, r_{\mathcal{R}}) \right\rangle : \\ \mathcal{V}^\tau(\mathsf{com}, \mathsf{decom}) = 0, \mathcal{V}^\tau(\mathsf{com}, \mathsf{decom}') = 1 \end{array} \right] > 1 - \frac{1}{p(n)}$$

for all sufficiently large n, where $\mathsf{depth}(n) = \left\lceil \frac{c(n)}{\log n} \right\rceil + 1$.

We note that the above theorem holds even if the commitment scheme is statistically-hiding only against an honest receiver. In what follows we introduce the notation used in this section. We proceed with a brief presentation of the main ideas underlying the proof of Theorem 3.3, which is then followed by a formal description of the malicious sender $\mathcal{S}^*$.

Notations. Let $(\mathcal{S}, \mathcal{R})$ be a bit-commitment scheme with oracle access to a family of trapdoor permutations. We denote by $b \in \{0,1\}$ and $r_{\mathcal{S}}, r_{\mathcal{R}} \in \{0,1\}^*$ the input bit of the sender and the random coins of the sender and the receiver, respectively. We denote by $c(n)$ the maximal number of bits communicated from the sender to the receiver in the commit stage with security parameter 1^n. In addition we denote by $d(n)$ the number of communication rounds in the scheme with security parameter 1^n, and without loss of generality we assume that the receiver makes the first move. Each communication round consists of a message sent from the receiver to the sender followed by a message sent from the sender to the receiver. We denote by q_i and a_i the messages sent by the receiver and the sender in the i-th round, respectively, and denote by a_{d+1} the message sent by the sender in the reveal stage. Finally, we let $\bar{a}_i = (a_1, \ldots, a_i)$ and $\bar{q}_i = (q_1, \ldots, q_i)$.

Although the sender is a probabilistic polynomial-time *Turing-machine*, in order to interact with the oracle Sam we need to identify the sender with a sequence of polynomial-size *circuits* $S_1, \ldots, S_{d+1}$ as follows. In the first round, $\mathcal{S}$ sends a_1 by computing $a_1 = S_1(b, r_{\mathcal{S}}, q_1)$. Similarly, in the following rounds, $\mathcal{S}$ sends a_i by computing $a_i = S_i(b, r_{\mathcal{S}}, \bar{q}_i)$.

Finally, in order to simplify the notation regarding the input and output of the oracle Sam, in this section we ignore parts of the input and output of Sam: we ignore the security parameter and the "signatures" (since our malicious sender $\mathcal{S}^*$ will only ask legal queries), and consider queries of a simplified form $Q = (C^{\tau}_{\text{next}}, C^{\tau}, z)$, and answers that consist only of w' (i.e., an answer consists only of a uniformly distributed preimage of z under the mapping defined by C^{τ}). In addition, in what follows it will be more intuitive to replace z in the queries by its preimage w, but this is clearly not essential.

A brief overview. Informally, recall that the oracle Sam described in Section 3.1 acts as follows: Sam is given as input a query $Q = (C_{\text{next}}, C, z)$, and outputs a pair (w', z') where w' is a uniformly distributed preimage of z under the mapping defined by the circuit C, and $z' = C_{\text{next}}(w')$. In addition, we imposed the restriction that there was a previous query $(C, \cdot, \cdot)$ that was answered by (w, z) (note that this imposes a forest-like structure on the queries), and we only allow querying Sam up to depth $O(n/\log n)$.

Given a statistically-hiding bit-commitment scheme in which the sender communicates $c(n)$ bits during the commit stage, we assume without loss of generality that the commit stage of the scheme has $c(n)$ communication rounds, where in each round the sender communicates a single bit to the receiver. The malicious sender $\mathcal{S}^*$ operates as follows: it chooses a random input w (consisting of random coins and a random committed bit), and during the first $\log n$ rounds it simulates the honest sender. In these $\log n$ rounds, it receives $\log n$ messages $q_1, \ldots, q_{\log n}$ from the receiver. Then, $\mathcal{S}^*$ constructs the circuit $C_{q_1,\ldots,q_{\log n}}$ that receives as input the sender's input w and outputs the $\log n$ sender's messages corresponding to the receiver's messages $q_1, \ldots, q_{\log n}$. This circuit is used to query Sam for a random input w_1. It may be the case, however, that w_1 is not consistent with the actual messages $a_1, \ldots, a_{\log n}$ that $\mathcal{S}^*$ sent in the first $\log n$ rounds. In this case, $\mathcal{S}^*$ "rewinds" Sam for a polynomial number of times, and since the total length of the sender's messages in these $\log n$ rounds is only $\log n$ bits, then with sufficiently high probability $\mathcal{S}^*$ will obtain a consistent w_1. Now, in the next $\log n$ rounds the malicious sender $\mathcal{S}^*$ simulates the honest sender with input w_1, and at the end of these $\log n$ rounds it will query (and rewind) Sam again for another consistent input $w_{\log n+1}$, and so on. Finally, after completing the commit stage, $\mathcal{S}^*$ queries Sam to obtain two random inputs $w_{c(n)}$ and $w'_{c(n)}$ which are consistent with the transcript of the commit stage. Since the commitment scheme is statistically-hiding, with probability roughly half they can be used to break the binding of the protocol. A crucial point in this description, is that $\mathcal{S}^*$ queries Sam only up to depth $c(n)/\log n$ ($\mathcal{S}^*$ used Sam to obtain $c(n)/\log n$ values $w_1, w_{\log n+1}, \ldots, w_{c(n)}$). Therefore, if $c(n) = o(n)$, then an oracle Sam that answers queries only up to depth $c(n)/\log n$ cannot be used to invert a random trapdoor permutation, according to Theorem 3.2.

A formal description of $\mathcal{S}^*$. Given a bit-commitment scheme $(\mathcal{S}, \mathcal{R})$ in which the sender communicates $c(n)$ bits during the commit stage, we assume without loss of generality (and for simplicity of the presentation) that the scheme has $c(n)$ communication rounds (i.e., $d(n) = c(n)$) where in each round during the commit stage the sender communicates a single bit to the receiver (i.e., each of $a_1, \ldots, a_{d(n)}$ is one bit). Furthermore, in order to simplify the description of $\mathcal{S}^*$, we assume that $\log n$ is an integral value (where 1^n is the security parameter given as input to $\mathcal{S}^*$) and that $c(n) = M \cdot \log n + 1$ for some integer $M = M(n)$. We stress that these assumptions are not at all essential, but avoiding them will result in a more complicated description. On input 1^n, the malicious sender $\mathcal{S}^*$ with oracle access to $\mathsf{Sam}_{\mathsf{depth}}^{\tau,\mathcal{F},\mathsf{sign}}$ interacts with the honest receiver $\mathcal{R}$ as follows.

1. **The commit stage:**
   (a) In the first round $\mathcal{S}^*$ receives $\mathcal{R}$'s message q_1, and computes the description of the circuit $C_1 = S_1(\cdot, \cdot, q_1)$ obtained from the circuit S_1 by fixing q_1 as its third input. Then, $\mathcal{S}^*$ queries $\mathsf{Sam}_{\mathsf{depth}}^{\tau,\mathcal{F},\mathsf{sign}}$ with $(C_1, \perp, \perp)$, receives an answer $w_1 = (b_1, r_1)$ and sends $a_1 = S_1(b_1, r_1, q_1)$ to $\mathcal{R}$.
   (b) In every round $i \in \{2, \ldots, \log n\}$, $\mathcal{S}^*$ simulates the honest sender $\mathcal{S}$ with input w_1. That is, $\mathcal{S}^*$ receives $\mathcal{R}$'s message q_i and replies with $a_i = S_i(b_1, r_1, \bar{q}_i)$.
   (c) In round $\log n + 1$, $\mathcal{S}^*$ receives $\mathcal{R}$'s message $q_{\log n+1}$, and computes the description of the circuit $C_{\log n+1} = S_{\log n+1}(\cdot, \cdot, \bar{q}_{\log n+1})$ obtained from the circuit $S_{\log n+1}$ by fixing $\bar{q}_{\log n+1}$ as its third input. Then, $\mathcal{S}^*$ queries $\mathsf{Sam}_{\mathsf{depth}}^{\tau,\mathcal{F},\mathsf{sign}}$ with $(C_{\log n+1}, C_1, w_1)$ for $t = 2n^5 c(n)p(n)$ times and receives t answers. If one of these answers is consistent with the transcript of the protocol so far, then denote the first such answer by $w_{\log n+1} = (b_{\log n+1}, r_{\log n+1})$, and in this case $\mathcal{S}^*$ sends the message $a_{\log n+1} = S_{\log n+1}(b_{\log n+1}, r_{\log n+1}, \bar{q}_{\log n+1})$ to $\mathcal{R}$. Otherwise, $\mathcal{S}^*$ aborts the execution of the protocol.
   (d) In the remainder of the commit stage $\mathcal{S}^*$ acts as follows:
      i. For every k and in every round $i \in \{(k-1)\log n + 2, \ldots, k \log n\}$, the malicious sender $\mathcal{S}^*$ simulates the honest sender $\mathcal{S}$ with input $w_{(k-1)\log n+1}$.
      ii. For every integer k and in every round $k \log n + 1$ the malicious sender $\mathcal{S}^*$ receives $\mathcal{R}$'s message $q_{k \log n+1}$, and computes the description of the circuit $C_{k \log n+1} = S_{k \log n+1}(\cdot, \cdot, \bar{q}_{k \log n+1})$ obtained from the circuit $S_{k \log n+1}$ by fixing $\bar{q}_{k \log n+1}$ as its third input. Then, $\mathcal{S}^*$ queries $\mathsf{Sam}_{\mathsf{depth}}^{\tau,\mathcal{F},\mathsf{sign}}$ with $(C_{k \log n+1}, C_{(k-1)\log n+1}, w_{(k-1)\log n+1})$ for $t = 2n^5 c(n)p(n)$ times and receives t answers. If one of these answers is consistent with the transcript of the protocol so far, then denote the first such answer by $w_{k \log n+1} = (b_{k \log n+1}, r_{k \log n+1})$, and in this case $\mathcal{S}^*$ sends $a_{k \log n+1} = S_{k \log n+1}(b_{k \log n+1}, r_{k \log n+1}, \bar{q}_{k \log n+1})$ to $\mathcal{R}$. Otherwise, $\mathcal{S}^*$ aborts the execution of the protocol.

2. **The reveal stage:**
   (a) $\mathcal{S}^*$ queries $\mathsf{Sam}^{\tau,\mathcal{F},\mathsf{sign}}_{\mathsf{depth}}$ with $(\perp, C_{d(n)}, w_{d(n)})$ for n times, and receives n pairs $\left\{\left(b^{(j)}_{d(n)+1}, r^{(j)}_{d(n)+1}\right)\right\}^{n}_{j=1}$. If there exist $j_0, j_1 \in [n]$ such that $b^{(j_0)}_{d(n)+1} = 0$ and $b^{(j_1)}_{d(n)+1} = 1$, then $\mathcal{S}^*$ outputs the two values

$$\begin{aligned}\mathsf{decom} &= S_{d(n)+1}\left(b^{(j_0)}_{d(n)+1}, r^{(j_0)}_{d(n)+1}, \bar{q}_{d(n)}\right)\\ \mathsf{decom}' &= S_{d(n)+1}\left(b^{(j_1)}_{d(n)+1}, r^{(j_1)}_{d(n)+1}, \bar{q}_{d(n)}\right) \quad .\end{aligned}$$

   Otherwise, $\mathcal{S}^*$ aborts the execution of the protocol.

Two minor technical details were omitted from the description. First, according to the description of Sam (Section 3.1), whenever Sam is queried multiple times with the same input, it returns the exact same answer. Thus, whenever $\mathcal{S}^*$ queries Sam more than once with the same input, $\mathcal{S}^*$ has to make sure that the queries are all different (for example, by artificially embedding the query number to one of the circuits in the query). Second, in order for $\mathcal{S}^*$'s queries to be legal, it should hold that the circuit $C_{k\log n+1}$ is a refinement of the circuit $C_{(k-1)\log n+1}$ for every integer k (as discussed in Section 3.1). This can be done very easily by embedding the description of each $C_{(k-1)\log n+1}$ inside each $C_{k\log n+1}$ (i.e., the output of C_i is the sequence of bits $\bar{a}_i$ instead of only the bit a_i).

The formal proof proceeds by arguing that $\mathcal{S}^*$ successfully completes the commit stage with high probability. Then, given that $\mathcal{S}^*$ has successfully completed the commit stage, we prove that the transcript of the commit stage is distributed identically to the transcript of the commit stage in an honest execution of the protocol. This enables us to use the fact that the commitment scheme is statistically-hiding, and therefore a random transcript can be revealed both as a commitment to $b = 0$ and as a commitment to $b = 1$, with almost equal probabilities. Due to space limitations we refer the reader to [19] for a formal proof, which then immediately implies the correctness of Theorem 3.1.

4 Refining the Relation Between Single-Server PIR and Commitment Schemes

The relation between single-server PIR and commitment schemes was first explored by Beimel et al. [1], who showed that any single-server PIR protocol in which the server communicates at most $n/2$ bits to the user (where n is the size of the server's database), can be used to construct a weakly-binding statistically-hiding bit-commitment scheme. In particular, this served as the first indication that the existence of low-communication PIR protocols implies the existence of one-way functions. In this section, we refine the relation between these two fundamental primitives by improving their reduction. Informally speaking, our reduction essentially uses the reduction of Beimel et al. instantiated with a better extractor. This enables the following improvements: (1) the communication complexity of the PIR protocol is essentially preserved, (2) given a single-server PIR protocol in which the server communicates $n - k$ bits, it is possible to commit to $\Omega(k)$ bits while executing the underlying single-server PIR protocol only once, and (3) whereas the construction of Beimel et al. was presented for single-server PIR protocols in which the server communicates at most $n/2$ bits, our construction can rely on single-server PIR in which the server communicates up to $n - \omega(\log n)$ bits.

In what follows we state our main theorem in the current section, and then turn to formally describe the construction and to provide intuition for its proof. Due to space limitations we refer the reader to [19] for the formal proof.

Theorem 4.1. *Let $d(n) \in \omega(\log n)$, $k(n) \geq 2d(n)$, and let $\mathcal{P}$ be a single-server PIR protocol in which the server communicates $n - k(n)$ bits, where n is the size of the server's database. Then, there exists a weakly-binding statistically-hiding commitment scheme $\mathcal{COM}^{\mathcal{P}}$ for $d(n)/6$ bits, in which the sender communicates less than $n - k(n) + 2d(n)$ bits during the commit stage. Moreover, the construction is fully-black-box and linearly-preserving.*

The construction. Fix $d(n)$, $k(n)$ and $\mathcal{P}$ as in Theorem 4.1. In the construction we use a strong $\left(d(n)/3, 2^{1-d(n)/3}\right)$-extractor EXT : $\{0,1\}^n \times \{0,1\}^{d(n)} \rightarrow \{0,1\}^{d(n)/6}$ whose existence is guaranteed by Proposition 2.1. Figure 4 describes our construction of the commitment scheme $\mathcal{COM}^{\mathcal{P}} = (\mathcal{S}, \mathcal{R})$. The correctness of $\mathcal{COM}^{\mathcal{P}}$ follows directly from the correctness of $\mathcal{P}$. In addition, notice that the total number of bits communicated by the sender in the commit stage is the total number of bits that the server communicates in $\mathcal{P}$ plus the seed length and the output length of the extractor EXT. Thus, the sender communicates less than $n - k(n) + 2d(n)$ bits during the commit stage.

Proof intuition. The commit stage consists of the sender and the receiver choosing random inputs $x \in \{0,1\}^n$ and $i \in [n]$, respectively, and executing the PIR protocol $\mathcal{P}$ on these inputs. As a consequence, the receiver obtains a bit x_i, which by the correctness of $\mathcal{P}$ is the i^{th} bit of x. Now, notice that since the sender communicated only $n - \omega(\log n)$ bits, then the random variable corresponding to x still has $\omega(\log n)$ min-entropy from the receiver's point of view (with high probability). We take advantage of this fact, and exploit the remaining min-entropy of x in order to hide the committed string s in a statistical manner (note that since it is required to reveal the seed of the extractor during the commit stage, we need a *strong* extractor). The formal proof of the hiding property is similar to that of Lu [29] in the bounded storage model, which is in turn based on ideas that were used for constructing pseudorandom generators for space bounded computations [33]. We note that the proof of hiding does not rely on any computational properties of the underlying PIR protocol $\mathcal{P}$, but only on the assumed bound on the number of bits communicated by the server in $\mathcal{P}$. The binding property follows from the security of the PIR protocol: in the reveal stage, the sender must send a value x whose i^{th} bit is consistent with the bit obtained by the receiver during the commit stage – but this bit is not known to the sender.

Protocol $\mathcal{COM}^{\mathcal{P}} = (\mathcal{S}, \mathcal{R})$

Joint input: security parameter 1^n.
Sender's input: $s \in \{0,1\}^{d(n)/6}$.

Commit stage:

1. $\mathcal{S}$ chooses a uniformly distributed $x \in \{0,1\}^n$.
2. $\mathcal{R}$ chooses a uniformly distributed index $i \in [n]$.
3. $\mathcal{S}$ and $\mathcal{R}$ execute the single-server PIR protocol $\mathcal{P}$ for database of length n, where $\mathcal{S}$ acts as the server with input x and $\mathcal{R}$ acts as the user with input i. As a result, $\mathcal{R}$ obtains a bit $x_i \in \{0,1\}$.
4. $\mathcal{S}$ chooses a uniformly distributed seed $t \in \{0,1\}^{d(n)}$, computes $y = \mathrm{EXT}(x,t) \oplus s$, and sends (t,y) to $\mathcal{R}$.

Reveal stage:

1. $\mathcal{S}$ sends (s,x) to $\mathcal{R}$.
2. If the i^{th} bit of x equals x_i and $y = \mathrm{EXT}(x,t) \oplus s$, then $\mathcal{R}$ outputs s. Otherwise, $\mathcal{R}$ outputs $\perp$.

Fig. 2. A construction of a commitment scheme from any low-communication single-server PIR protocol

5 Concluding Remarks

Our result does not rule out weakly-preserving (fully-black-box) constructions of single-server PIR from trapdoor permutations in which the sender communicates $o(n)$ bits to the user. We note that although weakly-preserving reductions guarantee much weaker security than polynomially-preserving reductions, investigating lower bounds for such reductions is still a very interesting research topic. Even more so as the sole construction to date of a single-server PIR protocol from trapdoor permutations uses such a reduction. A possible step towards tightening our bound is to first provide an improved lower bound on the communication complexity of statistically-hiding commitment schemes that allow the sender to commit to more than a single bit. Whereas in Section 4 we proved that any low-communication single-server PIR implies a statistically-hiding commitment scheme that allows the sender to commit to a relatively long string, our lower bound on the communication complexity of statistically-hiding commitment schemes in Section 3 serves as a bottleneck: it does not take into consideration the number of committed bits (the lower bound is only in terms of the security parameter).

It is quite possible that a much tighter lower bound can be proved for string-commitment schemes. Such a lower bound may extend the result of the current paper to the setting of weakly-preserving reductions, and prove the optimality of the single-server PIR protocol of Kushilevitz and Ostrovsky [27]. We note that the statistically-hiding commitment scheme of Naor et al. [31] (which is constructed from one-way permutations in a fully-black-box manner) can be used to commit to $O(\log n)$ bits while the sender communicates $O(n)$ bits (see, for example, [32]).

Acknowledgments. We are grateful to Yuval Ishai and Omer Reingold for many useful conversations and observations. We also thank the anonymous referees for their remarks and suggestions.

References

1. Beimel, A., Ishai, Y., Kushilevitz, E., Malkin, T.: One-way functions are essential for single-server private information retrieval. In: 31st STOC, pp. 89–98 (1999)
2. Cachin, C., Micali, S., Stadler, M.: Computationally private information retrieval with polylogarithmic communication. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 402–414. Springer, Heidelberg (1999)
3. Chang, Y.: Single database private information retrieval with logarithmic communication. In: 9th ACISP, pp. 50–61 (2004)
4. Chor, B., Goldreich, O., Kushilevitz, E., Sudan, M.: Private information retrieval. In: 36th FOCS, pp. 41–50 (1995)
5. Di Crescenzo, G., Malkin, T., Ostrovsky, R.: Single database private information retrieval implies oblivious transfer. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 122–138. Springer, Heidelberg (2000)
6. Dziembowski, S., Maurer, U.M.: On generating the initial key in the bounded-storage model. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 126–137. Springer, Heidelberg (2004)
7. Fischlin, M.: On the impossibility of constructing non-interactive statistically-secret protocols from any trapdoor one-way function. In: CT-RSA, pp. 79–95 (2002)
8. Gennaro, R., Gertner, Y., Katz, J.: Lower bounds on the efficiency of encryption and digital signature schemes. In: 35th STOC, pp. 417–425 (2003)
9. Gennaro, R., Gertner, Y., Katz, J., Trevisan, L.: Bounds on the efficiency of generic cryptographic constructions. SIAM J. Comput. 35(1), 217–246 (2005)
10. Gennaro, R., Lindell, Y., Malkin, T.: Enhanced versus plain trapdoor permutations for non-interactive zero-knowledge and oblivious transfer. Manuscript (2006)
11. Gennaro, R., Trevisan, L.: Lower bounds on the efficiency of generic cryptographic constructions. In: 41st FOCS, pp. 305–313 (2000)
12. Gentry, C., Ramzan, Z.: Single-database private information retrieval with constant communication rate. In: 32nd ICALP, pp. 803–815 (2005)
13. Gertner, Y., Kannan, S., Malkin, T., Reingold, O., Viswanathan, M.: The relationship between public key encryption and oblivious transfer. In: 41st FOCS, pp. 325–335 (2000)
14. Gertner, Y., Malkin, T., Reingold, O.: On the impossibility of basing trapdoor functions on trapdoor predicates. In: 42nd FOCS, pp. 126–135 (2001)
15. Goldreich, O.: Foundations of Cryptography, Basic Tools, vol. 1. Cambridge University Press, Cambridge (2001)
16. Goldreich, O.: Foundations of Cryptography, Basic Applications, vol. 2. Cambridge University Press, Cambridge (2004)
17. Haitner, I.: Implementing oblivious transfer using collection of dense trapdoor permutations. In: 1st TCC, pp. 394–409 (2004)
18. Haitner, I., Hoch, J.J., Reingold, O., Segev, G.: Finding collisions in interactive protocols – A tight lower bound on the round complexity of statistically-hiding commitments. In: 48th FOCS, pp. 669–679 (2007)
19. Haitner, I., Hoch, J.J., Segev, G.: A linear lower bound on the communication complexity of single-server private information retrieval. Cryptology ePrint Archive, Report 2007/351 (2007)

20. Harnik, D., Naor, M.: On the compressibility of NP instances and cryptographic applications. In: 47th FOCS, pp. 719–728 (2006)
21. Horvitz, O., Katz, J.: Bounds on the efficiency of "black-box" commitment schemes. In: 32nd ICALP, pp. 128–139 (2005)
22. Impagliazzo, R., Rudich, S.: Limits on the provable consequences of one-way permutations. In: 21st STOC, pp. 44–61 (1989)
23. Ishai, Y., Kushilevitz, E., Ostrovsky, R.: Sufficient conditions for collision-resistant hashing. In: 2nd TCC, pp. 445–456 (2005)
24. Kalai, Y.T., Raz, R.: Succinct non-interactive zero-knowledge proofs with preprocessing for LOGSNP. In: 47th FOCS, pp. 355–366 (2006)
25. Kim, J.H., Simon, D.R., Tetali, P.: Limits on the efficiency of one-way permutation-based hash functions. In: 40th FOCS, pp. 535–542 (1999)
26. Kushilevitz, E., Ostrovsky, R.: Replication is NOT needed: SINGLE database, computationally-private information retrieval. In: 38th FOCS, pp. 364–373 (1997)
27. Kushilevitz, E., Ostrovsky, R.: One-way trapdoor permutations are sufficient for non-trivial single-server private information retrieval. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 104–121. Springer, Heidelberg (2000)
28. Lipmaa, H.: An oblivious transfer protocol with log-squared communication. In: 8th ISC, pp. 314–328 (2005)
29. Lu, C.-J.: Encryption against storage-bounded adversaries from on-line strong extractors. J. Cryptology 17(1), 27–42 (2004)
30. Luby, M.: Pseudorandomness and Cryptographic Applications. Princeton University Press, Princeton (1996)
31. Naor, M., Ostrovsky, R., Venkatesan, R., Yung, M.: Perfect zero-knowledge arguments for NP using any one-way permutation. J. Cryptology 11(2), 87–108 (1998)
32. Nguyen, M.-H., Ong, S.J., Vadhan, S.P.: Statistical zero-knowledge arguments for NP from any one-way function. In: 47th FOCS, pp. 3–14 (2006)
33. Nisan, N., Zuckerman, D.: Randomness is linear in space. Journal of Computer and System Sciences 52(1), 43–52 (1996)
34. Ostrovsky, R., Skeith, W.E.: Algebraic lower bounds for computing on encrypted data. Cryptology ePrint Archive, Report 2007/064 (2007)
35. Ostrovsky, R., Skeith, W.E.: A survey of single database PIR: Techniques and applications. Cryptology ePrint Archive, Report 2007/059 (2007)
36. Reingold, O., Trevisan, L., Vadhan, S.P.: Notions of reducibility between cryptographic primitives. In: 1st TCC, pp. 1–20 (2004)
37. Rudich, S.: Limits on the provable consequences of one-way functions. PhD thesis, EECS Department, University of California, Berkeley (1988)
38. Simon, D.R.: Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 334–345. Springer, Heidelberg (1998)
39. Srinivasan, A., Zuckerman, D.: Computing with very weak random sources. SIAM J. Comput. 28(4), 1433–1459 (1999)
40. Stern, J.P.: A new efficient all-or-nothing disclosure of secrets protocol. In: Ohta, K., Pei, D. (eds.) ASIACRYPT 1998. LNCS, vol. 1514, pp. 357–371. Springer, Heidelberg (1998)
41. Wee, H.: One-way permutations, interactive hashing and statistically hiding commitments. In: 4th TCC, pp. 419–433 (2007)